AMENDMENTS TO THE CLAIMS

Claims 2–42 are pending in this application. As of this Response, Applicant has amended Claims 2, 10, 17, 31 and 37. Claims 3–9, 11–16, 18–30, 32–36 and 38–42 remain as previously pending.

1. (Canceled)

2. (Currently Amended) A method for providing data transfers between a processor and a component, the method comprising:

buffering an address with a first buffer, the first buffer being in communication with a processor and a component, wherein the processor operates at a different speed than the component;

buffering a data value with a second buffer, the second buffer being in communication with the processor and the component;

controlling the first buffer and the second buffer as a matched pair such that the address held in the first buffer corresponds to the data value held in the second buffer; and

controlling bi-directional data flow through the second buffer such that data flows between the processor and the component.

3. (Previously Presented) The method of Claim 2, wherein the first and second buffers are in communication with the processor via a bus.

4. (Previously Presented) The method of Claim 3, wherein the first and second buffers are in communication with the bus via a bus master controller and a bus slave controller.

5. (Previously Presented) The method of Claim 2, wherein the first buffer further comprises status bits.

6. (Previously Presented) The method of Claim 5, wherein the status bits relate to the type of request being made by the processor.

7. (Previously Presented) The method of Claim 2, wherein said act of controlling the first buffer and the second buffer as a matched pair is performed with pointers.

8. (Previously Presented) The method of Claim 2, wherein the processor is from an Intel Pentium® family of processors.

9. (Previously Presented) The method of Claim 2, wherein said act of controlling bi-directional data flow is performed with at least one input data arbiter.

10. (Currently Amended) A method for controlling data transfers between a processor and a component, the method comprising:

buffering with a plurality of address buffers address requests from a processor to a component, wherein the processor operates at a different speed than the component;

bi-directionally buffering with a plurality of data buffers data transfers between the processor and the component; and

controlling said buffering address requests and said bi-directionally buffering such that each of the buffered data transfers relates to an address held in one of the plurality of address buffers.

11. (Previously Presented) The method of Claim 10, additionally comprising indicating which of the plurality of data buffers is available to accept new data.

12. (Previously Presented) The method of Claim 11, wherein said act of indicating is performed with reference pointers.

13. (Previously Presented) The method of Claim 10, wherein the processor is from an Intel Pentium® family of processors.

14. (Previously Presented) The method of Claim 10, wherein said act of buffering address requests includes the use of an input arbiter and an output arbiter.

15. (Previously Presented) The method of Claim 10, wherein said act of bi-directionally buffering is performed with an input arbiter and an output arbiter.

16. (Previously Presented) The method of Claim 10, wherein the plurality of address buffers comprises at least three address buffers and wherein the plurality of data buffers comprises at least three data buffers.

17. (Currently Amended) A method for providing data transfers between a processor and a component, the method comprising:

buffering a first address buffer with a first address;

buffering a second address buffer with a second address;

buffering a first data buffer with a first data value;

buffering a second data buffer with a second data value;

controlling the first address buffer and the first data buffer as a matched pair such that the first address corresponds to the first data value; and

controlling bi-directional data flow through the first data buffer such that data flows between a processor and a component, wherein the first and second address buffers and the first and second data buffers are each in communication with the processor and the component, and wherein the processor operates at a different speed than the component.

18. (Previously Presented) The method of Claim 17, wherein the processor is from an Intel Pentium® family of processors.

19. (Previously Presented) The method of Claim 17, wherein the first and second address buffers and the first and second data buffers are in communication with the processor via a bus.

20. (Previously Presented) The method of Claim 17, wherein said act of controlling the first address buffer and the first data buffer as a matched pair is performed with pointers.

21. (Previously Presented) The method of Claim 17, wherein said act of controlling bi-directional data flow is performed with at least one input data arbiter.

22. (Previously Presented) The method of Claim 17, wherein said act of controlling the first address buffer and the first data buffer as a matched pair allows data to be read from the first data buffer while an address is written to the first address buffer.

23. (Previously Presented) A method for transferring addresses and data through a bi-directional buffer, the method comprising:

storing an address in a first buffer in communication with a first component and a second component, the first buffer including status bits;

storing data in a second buffer matched with said first buffer so that the address stored in the first buffer is related to the data stored in the second buffer; and

providing signals with an arbiter in communication with said status bits so as to grant access to the first buffer and to the second buffer such that a second address can be written to the first buffer while data is read from the second buffer.

24. (Previously Presented) The method of Claim 23, wherein the status bits comprise transfer type bits indicative of the status of an address transfer from the first component to the first buffer.

25. (Previously Presented) The method of Claim 23, wherein the status bits comprise transfer type bits indicative of the status of a data transfer from the first component to the second buffer.

26. (Previously Presented) The method of Claim 23, wherein the first component comprises a memory.

27. (Previously Presented) The method of Claim 23, wherein the first component comprises a processor.

28. (Previously Presented) The method of Claim 27, wherein the first buffer is in communication with the processor via a bus.

29. (Previously Presented) The method of Claim 28, wherein the first buffer is in communication with the bus via a bus master controller and a bus slave controller.

30. (Previously Presented) The method of Claim 27, wherein the processor is from an Intel Pentium® family of processors.

31. (Currently Amended) A method for transferring data between a processor and a component utilizing a plurality of address buffers and a plurality of data buffers, the method comprising:

receiving a data request including an associated address from a processor;

determining whether an address buffer and an associated data buffer are available;

storing the associated address in a first address buffer;

transmitting with a first bi-directional data buffer associated with the first address buffer a data request including said associated address to a component; and

receiving data from the component in the first bi-directional data buffer.

32. (Previously Presented) The method of Claim 31, additionally comprising receiving the address into the first address buffer while data is being read from the first data buffer.

33. (Previously Presented) The method of Claim 31, wherein the first address buffer and the first data buffer are in communication with the processor via a bus.

34. (Previously Presented) The method of Claim 33, wherein the first address buffer and the first data buffer are in communication with the bus via a bus master controller and a bus slave controller.

35. (Previously Presented) The method of Claim 31, wherein the first data buffer and the first address buffer are associated with each other through the use of pointers.

36. (Previously Presented) The method of Claim 31, wherein the processor is from an Intel Pentium® family of processors.

37. (Previously Presented) An apparatus for controlling data transfers between a processor and a component, the apparatus comprising:

means for buffering address requests from a processor to a component, wherein the processor operates at a different speed than the component;

means for bi-directionally buffering data transfers between the processor and the component; and

control logic for controlling the means for buffering and the means for bi-directionally buffering so that each of the buffered data transfers relates to an address held in the means for buffering.

38. (Previously Presented) The apparatus of Claim 37, wherein the processor is from an Intel Pentium® family of processors.

39. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes a plurality of address buffers.

40. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes a plurality of data buffers.

41. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes an input arbiter and an output arbiter.

42. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes an input arbiter and an output arbiter.